What is claimed is:

1.   A method for validating network transformation instructions, the method comprising:

(a)   determining whether a plurality of network transformation instructions arranged in a first order will result in an invalid network state of a computer network if implemented; and

(b)   if the plurality of network transformation instructions arranged in the first order will result in an invalid network state if implemented, rearranging the plurality of network transformation instructions into an order that will not result in an invalid network state if implemented.

2.   The invention of Claim 1, wherein (b) comprises:

(b1)   rearranging the plurality of network transformation instructions; and

(b2)   determining whether the plurality of network transformation instructions in the re-arranged order will result in an invalid network state if implemented;

wherein (b1) and (b2) are repeated until the plurality of network transformation instructions will not result in an invalid network state if implemented.

3.   The invention of Claim 1, wherein an invalid network state results if the computer network is destabilized.

4.   The invention of Claim 1, wherein an invalid network state results if the computer network is partitioned.

5.   The invention of Claim 1, wherein an invalid network state results if a service level agreement is violated.

6.   The invention of Claim 1, wherein an invalid network state results if there is an undesirable intermediate state in the computer network.

7.   The invention of Claim 1, wherein an invalid network state results if the plurality of network transformation instructions cannot be implemented completely.

8.   The invention of Claim 1, wherein an invalid network state results if traffic demands are not met.

9.   The invention of Claim 1, wherein an invalid network state results if requested traffic characteristics are violated.

10.   The invention of Claim 1, wherein an invalid network state results if data must be dropped.

11.   The invention of Claim 1, wherein the act of determining in (a) is performed by a simulation program.

12.   The invention of Claim 1, wherein the act of determining in (a) is performed by a proof checker.

13.   The invention of Claim 12, wherein the proof checker uses a list of acceptable orders of network transformation instructions.

14.   The invention of Claim 1, wherein the act of determining in (a) is performed by a resolution theorem prover.

15.   The invention of Claim 14, wherein the resolution theorem prover uses as axioms a list of acceptable orders of network transformation instructions.

16.   The invention of Claim 14, wherein the resolution theorem prover comprises a list of acceptable orders of network transformation instructions.

17.   The invention of Claim 1, wherein the act of determining in (a) comprises comparing the plurality of network transformation instructions arranged in the first order against a list of acceptable orders of network transformation instructions.

18.   The invention of Claim 1 further comprising determining whether at least one network transformation instruction is valid.

19.   The invention of Claim 18, wherein the act of determining whether at least one network transformation instruction is valid is performed by a proof checker.

20.   The invention of Claim 19, wherein the proof checker comprises a list of acceptable network transformation instructions.

21.   The invention of Claim 18, wherein the act of determining whether at least one network transformation instruction is valid is performed by a resolution theorem prover.

22.   The invention of Claim 21, wherein the resolution theorem prover comprises a list of acceptable network transformation instructions.

23.   The invention of Claim 18, wherein the act of determining whether at least one network transformation instruction is valid comprises comparing the at least one network transformation instruction against a list of acceptable network transformation instructions.

24.   The invention of Claim 1 further comprising creating the plurality of network transformation instructions.

25.   The invention of Claim 1 further comprising:

(c)   implementing the plurality of network transformation instructions in the arrangement that will not result in an invalid network state.

26.   The invention of Claim 25, wherein (c) comprises configuring a plurality of network elements in the computer network using the plurality of network transformation instructions.

27.   The invention of Claim 26 further comprising converting the plurality of network transformation instructions into protocols appropriate for the plurality of network elements.

28.   A system for validating network transformation instructions, the system comprising:

a plurality of network elements in a computer network; and

a processor operative to determine whether a plurality of network transformation instructions arranged in a first order will result in an invalid network state of the computer network if implemented and further operative to rearrange the plurality of network transformation instructions into an order that will not result in an invalid network state if implemented.

29.   The invention of Claim 28, wherein the processor is further operative to repeatedly rearrange the plurality of network transformation instructions until an order is found that will not result in an invalid network state if implemented.

30. The invention of Claim 28, wherein an invalid network state results if the computer network is destabilized.

31. The invention of Claim 28, wherein an invalid network state results if the computer network is partitioned.

32. The invention of Claim 28, wherein an invalid network state results if a service level agreement is violated.

33. The invention of Claim 28, wherein an invalid network state results if there is an undesirable intermediate state in the computer network.

34. The invention of Claim 28, wherein an invalid network state results if the plurality of network transformation instructions cannot be implemented completely.

35. The invention of Claim 28, wherein an invalid network state results if traffic demands are not met.

36. The invention of Claim 28, wherein an invalid network state results if requested traffic characteristics are violated.

37. The invention of Claim 28, wherein an invalid network state results if data must be dropped.

38. The invention of Claim 28, wherein the processor is further operative to create the plurality of network transformation instructions.

39.   The invention of Claim 28, wherein the processor is further operative to implement the plurality of network transformation instructions in the arrangement that will not result in an invalid network state.

5

40.   A system for validating network transformation instructions, the system comprising:

means for determining whether a plurality of network transformation instructions arranged in a first order will result in an invalid network state of a computer network if implemented; and

10

means for, if the plurality of network transformation instructions arranged in the first order will result in an invalid network state if implemented, rearranging the plurality of network transformation instructions into an order that will not result in an invalid network state if implemented.

15